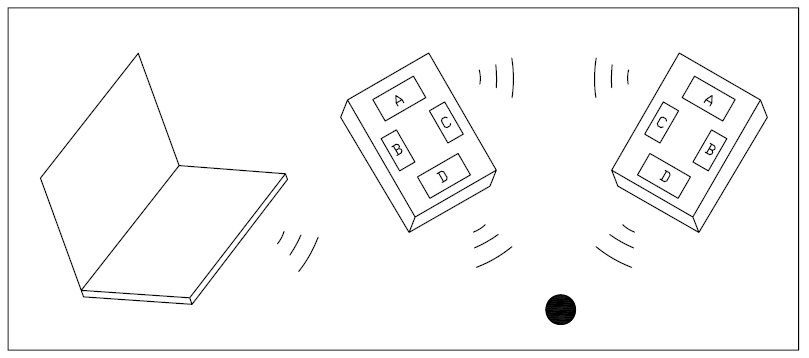
SYSTEM AND ENVIRONMENTAL CONSTRAINTS DOCUMENT



**Project:** Robots-That-Play-With-Balls

**Task:** Locating and moving a ball using two or more robots

**Document Version Number:** 02

**Date:** Oct 19, 2008

**Author:** Team Kilo

Contents

[SYSTEM AND ENVIRONMENTAL CONSTRAINTS DOCUMENT 1](#_Toc212284840)

[SYSTEM MODEL 3](#_Toc212284841)

[HARDWARE CONSTRAINTS 3](#_Toc212284842)

[SOFTWARE CONSTRAINTS 4](#_Toc212284843)

[ENVIRONMENTAL ISSUES 5](#_Toc212284844)

[COMPATIBILITY 5](#_Toc212284845)

[REUSABILITY 6](#_Toc212284846)

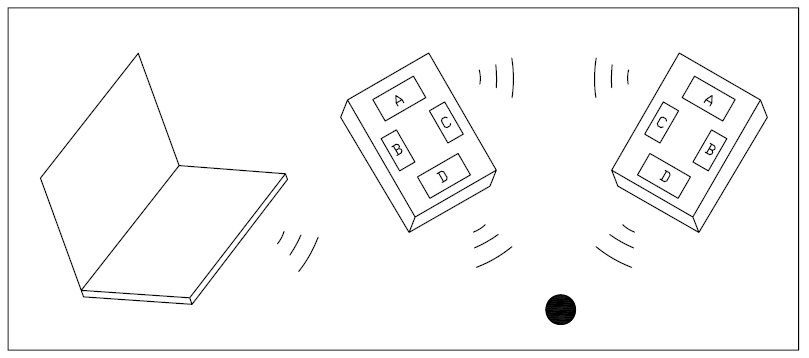
[STRUCTURES 6](#_Toc212284847)

[METHODOLOGIES 6](#_Toc212284848)

[TOOLS 7](#_Toc212284849)

[GLOSSARY OF TERMS 7](#_Toc212284850)

# SYSTEM MODEL



**Fig. 1 System model Diagram**

The functional system model diagram is as shown in Fig. 1. The working system is composed of two Robots and one (optional) laptop. Each Robot will have at least some of the following functionalities:

1. Bluetooth device
2. Light sensor
3. Compass sensor
4. Ultrasonic sensor

As described in (Client Requirements Document) CRD, these two Robots will work together to find a ball in an 4x4 m maze in which some obstacles are placed and transport the ball to a predefined location.

* Bluetooth is used for communicating between the Robots (and maybe the laptop).
* The compass sensor will be used to keep track of the Robot heading.
* Two light sensors will be used to detect the grid lines on the floor
* The ultrasonic sensor is used to detect obstacles around the robot
* A laptop might be used to coordinate with the Robots during the search. This option is subject to change as the project proceeds.

# HARDWARE CONSTRAINTS

**Mechanical Constraints**

* Sensor mounting

In this design, some sensors and one Bluetooth device are to be used for each Robot. The function and characteristics of each impose some constraints for the assembly. For instance, the compass sensor must be installed about 5’’ away from the driving wheels since the magnetic field around the motor may interfere with the built-in magnet in the compass sensor. The light sensor and US sensor must be installed in specific positions for them to function properly etc.

* End effecter

As described in the design document, the end effecter used to manipulate the ball must follow the design requirements; it has to work in such a way that the two Robots will co-operate to perform the task.

* Friction and slippery

Friction between the rolling wheels and the floor, friction between all the moving parts and stationary parts, slipping between the rolling wheels and travelling surface should be taken into account.

**System Constraints**

The response time of the system, the execution time of the code may also impose constraints on the main design.

# SOFTWARE CONSTRAINTS

NXC and NBC are two languages specially designed for NXT bricks. NXC stands for Not Exactly C which is similar to C. Compared to NBC which stands for NeXT Byte Code and is similar to assembler or machine language, NXC is a higher level language. NBC might be faster but the code is less readable and harder to maintain. These two languages could be combined for a more flexible design algorithm. NXC and NBC have the following characteristics:

**NXC:**

• Directly analogous to JAVA or C, generates NBC bytecode

• Gives reasonably direct access to the NXT controller through an API (application programming interface).

• Allows direct access to bytecode programming for those instances where the API

is missing functionality (ASM statement).

Compared to C, NXC has,

- No pointers or dynamic memory allocation

- Multi-dimensional arrays are implemented differently

- NXC has additional control structures and keywords e.g., the REPEAT statement, **task, start** keywords

**Using NBC: Pros**

• Complete access to NXT controller

• When speed matters

• Compactness (code size should be smaller than compiled code)

**Using NBC: Cons**

• No lexical scoping!! (All variables are GLOBAL)

• Control structures are more flexible

# ENVIRONMENTAL ISSUES

* **Operating environment**

Please refer to CRD 2.5 for details.

In addition, there will be 6 obstacles placed in the operating area- concrete cinder blocks measuring about 0.4m by about 0.15 m by about 0.2m high. These blocks will be in random orientations and may straddle the grid lines. The spacing between obstacles will be no less than 0.3 m.

There will be several red and blue balls placed in random locations around the competition area.

The picked ball will have to be deposited in a container positioned at a pre-specified location. The container will be a circular plastic bowl shaped object about 0.02m high and about 0.1m in diameter.

* **Operating requirement**

One of the Robots will be placed in the (0,0) position, and the other will be placed at (xmax,0), in this case, it will be (4000,0) where the coordinates are in mm. The original heading for these two Robots will be randomly chosen.

Timing is limited to 10 minutes.

# COMPATIBILITY

The code from previous labs will be subject to changes based on the followings:

* Each program was developed to perform certain specified task or tasks. When combined together, mutexes might need to be redesigned to avoid deadlock.
* Make sure all the global variables have distinct names

# REUSABILITY

The algorithm for odometer, Wall-follower, Lab 5 (Search and verify) and Lab 6 might be employed in the main design. As most of the code used in the labs is usually written in a modular way, a significant portion of it should be reusable, with little or no change.

The mounting for US sensor for Wall-follower and compass sensor might also be kept.

# STRUCTURES

(*Mechanical structures – reasons for choices, etc., electrical structures, design of software structures, etc. Some of this will derive from the constraints sections earlier. The rationale for choices must be given – this will allow a critical review of decisions, etc., before the design progresses too far or real physical systems are built.)*

**Note: These might be based on group discussions. As there are certain unknowns, we will not try to explore the design space before the meeting with the customer.**

# METHODOLOGIES

In terms of the overall approach to the problem, we will probably use the SCRUM methodology for development. This implies that we will have a couple of iterations over the design. We will generate a backlog of tasks at the kickoff meeting. We will start with a simple implementation that fulfills as many of the basic requirements as possible. At each iteration we will select a couple of tasks for that iteration and then complete them during the iteration. Of course, the overall design needs to be split into a couple of independent, modular components such that we can work on them in parallel.

* **General approach**

Option 1:

The two Robots will be working as a Master and a Slave respectively. First, the Master will start search for a ball in the maze and the slave will travel from the extreme x position to the origin, i.e. (0,0) position. If the Slave arrive at the origin and it will stop and pointing to 45 Degs ,i.e. pointing to the center of the maze and waiting for further instructions from the Master. If the master has found the ball, the Slave will follow the path that the master travels to meet it in the same grid. Then these two Robots will work together to transport the picked ball to the pre-defined location based on Bluetooth wireless communication.

Option 2:

The Master will find the ball first and instruct the Slave its current position and the Slave will travel to this position based on odometer. The Master will save, as it goes, the best path (shortest path around obstacles) and then communicate it to the Slave.\

Option 3:

Both robots will start searching at the same time (making sure not to interfere with each other). The first robot to find a ball will let the other one know what the position of the ball is. Then the second robot will find the way to the ball. At the same time, the first robot will find the way from the ball to the bowl. When both of these paths are known, the robots can start moving the ball.

There are more ideas, but we need more information, so we will wait until the meetings with the customer in order to clarify certain aspects.

* **Design algorithm**

1. Travel around in the maze

2. Distinguish obstacles and balls, or recognize a ball

3. Transport the ball

**Note: These will be based on group discussions**

# TOOLS

As described in 4.0, two languages are available for programming on these Robots. Please refer to section 4.0 for more details.

# GLOSSARY OF TERMS

CRD: Client requirement documentation which is an enclosed document with file No. 01